Proceedings of The 2016 IEEE International
Conference on Real-time Computing and Robotics
June 6-9, 2016, Angkor Wat, Cambodia

Finger-Eye: A Wearable Text Reading Assistive System for the Blind and Visually Impaired

Zhiming Liu, Yudong Luo, Jose Cordero, Na Zhao, and Yantao Shen*

Abstract— **This paper presents our recent research work in developing a portable and refreshable text reading system, called Finger-eye. In the system, a small camera is added to the fingertip-electrode interface of the current Electro-tactile Braille Display and placed on a blind person's finger to continuously process images using a developed rapid optical character recognition (OCR) method. This will allow translation of text to braille or audio with natural movement as if they were reading any Braille Display or book. The braille system that will be used is a portable electrical-based braille system that will eliminate the problems associated with refreshable mechanical braille displays. The goal of the research is to aid the blind and visually impaired (BVI) with a portable means to translate any text to braille, whether in the digital realm or physically, on any surface.**

I. Eye in the hand/Finger: A Finger-Eye System

Braille has constantly evolved to make life easier for the blind and visually impaired. From the inception of braille in 1824 to current day, braille has changed drastically and is widely available to the BVI community [1]. Refreshable braille displays are electro-mechanical bi-directional machines that allow for the translation of text to braille or braille to text [2]. These refreshable braille displays are state-of-the-art and are a giant leap in the evolution of braille. With the aid of a sighted person, they can upload information, such as, books, articles, newspapers, and even websites. However, there are still many limitations with modern braille equipment such as the complexity of use and physical size of the machine, which can limit the use of refreshable braille displays to younger persons and can be tedious to use and carry around [2].

In addition, existing technology allows the blind and visually impaired to download and translate various books and literature to braille or as audio [2]. However, there are many books and articles that do not have an audio or braille translation. This limits the resources that are available to the BVI and thus, their independence. This proposal will not only solve this problem to aid the BVI user.

A small camera will be added to the fingertip-electrode interface of the current Electrotactile Braille Display and placed on a blind person's finger to continuously process images for optical character recognition (OCR). This will allow translation of text to braille or audio with natural movement as if they were reading any Braille Display or book. The braille system that will be used is a portable electrical-based braille system that will eliminate the problems associated with refreshable mechanical braille displays. The goal of the research is to aid the BVI with a portable means to translate any text to braille, whether in the digital realm or physically, on any surface.

Current braille technology is limited, expensive and can be cumbersome to move around [3]. The technology that will be investigated includes developing an inexpensive, wearable light weight glove that contains the Electrotactile Braille Display and that can be placed on the hand of any user. The first step will be to ensure that the refreshable Electrotactile Braille Display is stable and performs with minimal error. The second step will be to test the effectiveness of the electric based refreshable Braille Display. This will require experiments that include sighted and BVI to test the effectiveness of electrical stimulation, and the effects of prolonged use. The third step will be designing a camera to be placed on the fingertip-electrode interface that would be fast enough to capture the quick reading speeds of the Braille reader. After the system is designed and developed, it will be tested with the BVI and their feedback will be used to improve the system and guide future adaptations to the Braille Display.

A. Electrotactile Braille Display

The first step of the project is to test the refreshable Electrotactile Braille Display. The system that will be used is sown in Figures 1, 2, and 3. The E-Braille Display, shown below, was developed at the University of Nevada Reno. Fig. 1 is the fingertip-electrode interface. This is where the Braille reader's finger will be placed for electrical stimulation. The lines coming out of the fingertip interface connect to the High Voltage converter (-300 volts to 300 volts), shown in Fig. 2, which has a fast slew rate (8 volts/μs) and is approximately linear with a gain of 53. The maximum allowable current of the op-amp is 10 mA which is more than enough for comfortable stimulation of the fingertip [4].

The input versus output voltage of the high voltage converter was plotted and the best fit line of this plot was used for the identification algorithm to reduce errors from the H.V converter. Fig. 3 shows the fingertip-electrode interface and the large return electrode modeled on a hand. The electrodes shown in Fig. 1 are sized and configured to standard (American Standard Sign) as shown in Fig. 4 [5].

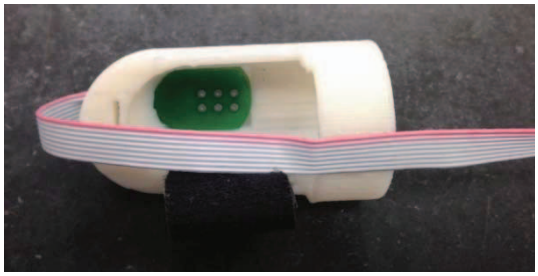


Fig. 1. Fingertip interface system.

Zhiming Liu, Yudong Luo, Jose Cordero, Na Zhao and Yantao Shen* are with the Department of Electrical and Biomedical Engineering, University of Nevada, Reno, 89557 USA (correspondence e-mail: ytshen@unr.edu).
The research work was partially supported by NSF CAREER Award CBET#1352006.

978-1-4673-8959-4/16/$31.00 © 2016 IEEE

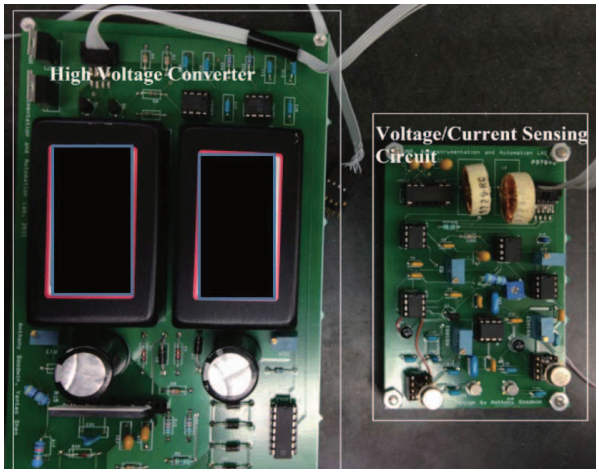


Fig. 2. High voltage converter and voltage/current sensing circuit.

Fig. 3: Fingertip and return electrode interface.

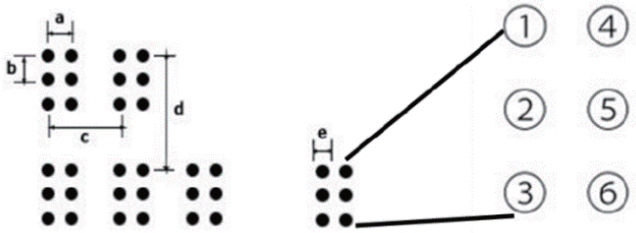


Fig 4. Standard US Braille cell dimensions (American Standard Sign, 2003).

B. *Finger-Eye System*

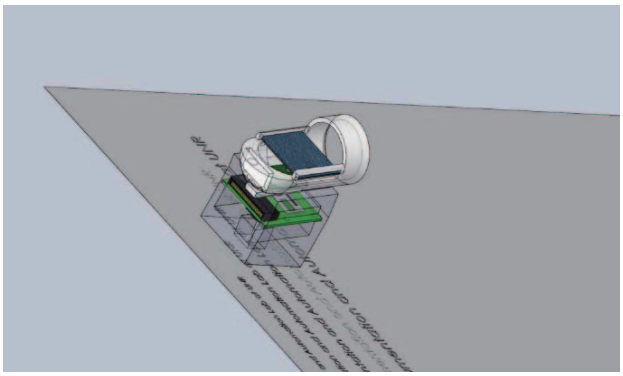


Fig. 5 3-D Finger-Eye model: The fingertip-electrode interface with a camera added for optical feedback.

The proposed Finger-Eye system is illustrated in Fig. 5. In the system, the fingertip-electrode interface with the added camera was designed by experimenting with the quality of the camera. It was found that the minimum distance between the surface and the camera. The vertical and horizontal angles of view for the camera were also considered in designing this preliminary sketch of the fingertip-electrode interface. The programming method that will be implemented will be designed to obtain maximum efficiency since the speed and memory of single board computer are limited. To achieve efficiency, preprocessing of the surface needs to done before applying an OCR algorithm. This will prevent the single board computer (SBC) from overheating, stalling, or crashing when an unexpected image is received.

The single board computer that will be used to implement the camera and braille system will be the Raspberry Pi, which has 512 MB of RAM and an SD card for storage [6]. The camera that will be used is an I2C camera (with CSI conversion) that is specifically designed for the Raspberry Pi.

One option for the Optical Character Recognition (OCR) algorithm that will be used is Tesseract-OCR which was developed and is maintained by Google [7]. The Electrotactile software and OCR algorithm will be implemented in a LINUX environment on the Raspberry Pi along with the GUI and an LCD for feedback and indication. The OCR engine is very popular and can be merged with other popular engines, such as the open source library of computer vision (OpenCV), to obtain higher character detection accuracy with low quality video stream and to calibrate and improve the system. In addition, another rapid OCR prototype method for the system is proposed and presented as below, which will be included in the current system in the future.

II. The Optical Character Recognition System

This section presents a new method for Optical Character Recognition (OCR) based on computer vision techniques. Our method relies on Pan-Tilt-Zoom (PTZ) camera to capture text image, because PTZ camera has the advantage of zooming in to capture high-resolution images, which definitely helps improve the accuracy of OCR. In experimental setup, a text page is placed on a planar table, which a PTZ camera looks at (see Fig. 6). A set of images are taken from the text page for OCR.

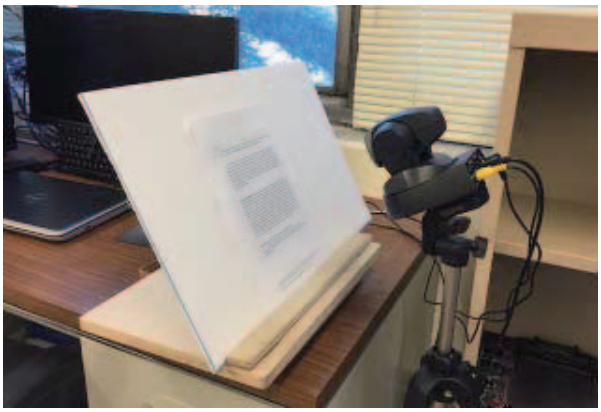


Fig. 6. OCR system setup.

A. *Image Preprocessing*

The lowest-resolution image is obtained by PTZ camera at zero zooming level. The image should include whole text region to be recognized. First, Gaussian smoothing is used for attenuating image noise. Second, the adaptive thresholding with Gaussian window partitions the image into objects (text) and background even there is large illumination variation in the image. Third, each connected component among objects is labeled and its shape descriptors such as bounding box and centroids are obtained.

B. *Estimation of Homography I*

Acquiring multiple images with different orientations by PTZ camera needs to pre-compute multiple pan-tilt angles. In our method, a chessboard attached to the planar table is first

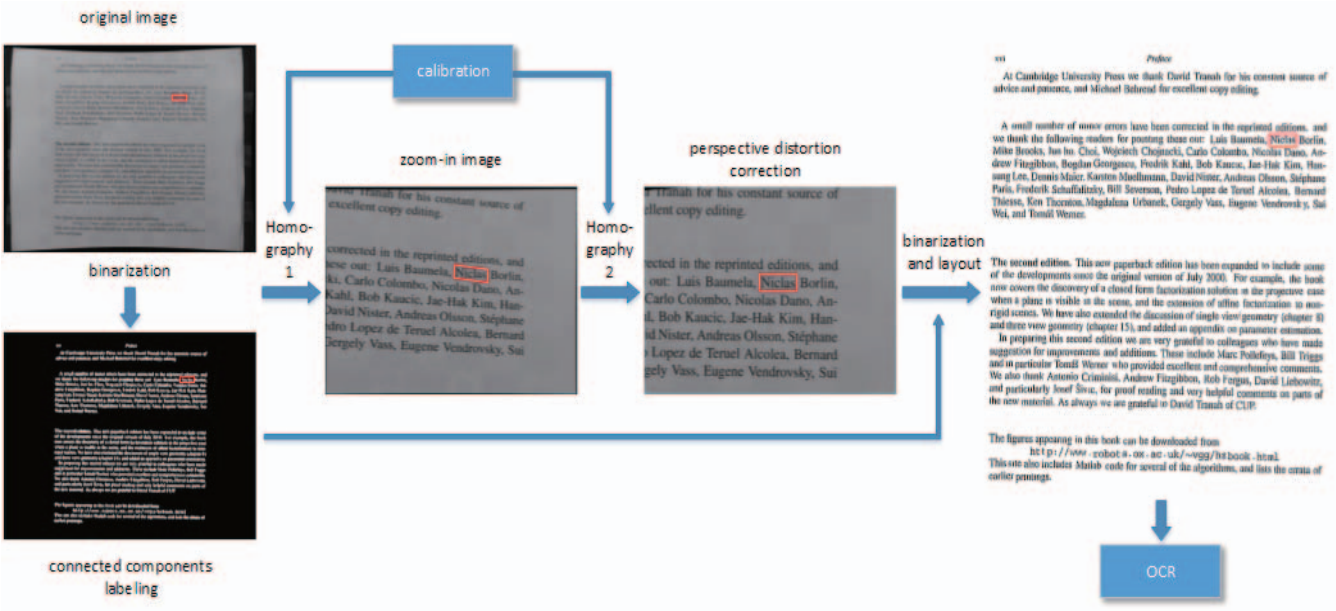


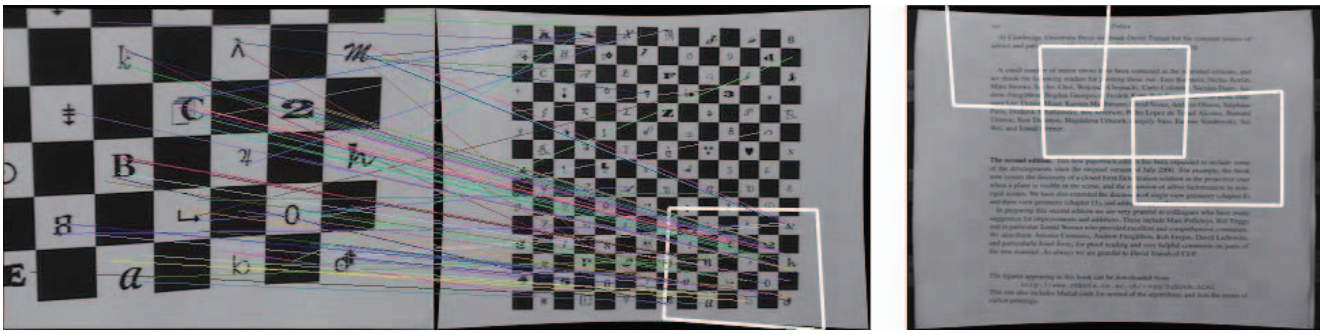
Fig. 7. Flowchart of OCR based on computer vision techniques.

Fig. 7 shows the flowchart of our method. First, the lowest-resolution image is obtained to contain whole text region within the image. Adaptive thresholding method is used to transform the grayscale image to a binary image. Connected components are then found and their locations and bounding boxes are recorded. Second, a set of zoom-in images are obtained to get closer and clearer views of text area by varying camera's pan-tilt-zoom. The bounding box of a word (connected component) in the lowest-resolution image is transformed via a homography to one of the zoom-in views, where the transformed word area should appear completely. Then the second homography is used to correct the perspective distortion of the image. Note that these two homographies are pre-computed in an offline calibration stage. Third, in the corrected image, a high-resolution word is first transformed to binary form. According to its position in the original image, a layout procedure then figures out where the binary form should appear in a large output image. This step repeatedly runs until each word has been processed. Last, Tesseract API [8] handles the resulting output image for OCR. The details of these stages are given in the following paragraphs.

used to estimate its relative position to camera. Then some anchor points on the chessboard are specified and their 3D positions in camera coordinate can be derived. If we assume that the center of camera rotation is fixed and coincides with the center of camera projection, the pan-tilt angles between these 3D points and camera center can be computed by geometry knowledge. A total of 18 anchor points on the chessboard are selected cautiously so as to the field of view of the resulting 18 zoom-in images can cover the whole area of the chessboard. Each zoom-in image and the whole chessboard image are related by a homography, see an example in Fig. 8(a).

If the camera's intrinsic parameters at different zoom levels can be accurately estimated and the rotation of camera in pan-tilt (i.e. external parameters) can be accurately controlled, the homography between any two images at different viewpoints and zooming can be computed by the following equation [9]:

$$x' = K'RK^{-1}x = H_1 x \quad (1)$$



(a) (b)

Fig. 8. Feature detection and matching for homography estimation. (a) The homography between a zoom-in image and the whole chessboard image. (b) Only three of 18 polygons are shown for the sake of clear display.

where H_1 is a projection transformation matrix, K' and K are intrinsic matrices with different zooming, R is the relative rotation matrix between the two views, x' and x are the images of the same object point.

However it is not easy to access the accurate PTZ camera's intrinsic and external parameters, we have to use the alternative method based on feature detection and matching for the accurate estimation of the homography H_1. Some randomly generated symbols are deliberately printed in the white squares on the chessboard to facilitate this process, but they do not affect the chessboard detection for camera calibration. Given a zoom-in image and the lowest-resolution image (the whole chessboard image), Scale-Invariant Feature Transform (SIFT) and fast approximate nearest neighbor search (FLANN) [10] are used for feature detection and matching between the two images. A perspective transformation a.k.a. homography between them is estimated with the use of RANSAC-based robust method.

Fig. 8(a) shows a found homography between two planar images. A white polygon is drawn in the whole chessboard image to indicate the region the zoom-in image can cover. As a result, 18 zoom-in images generate 18 homography matrices, which cast 18 polygon regions in the image. There should be a large overlapping area between any two adjacent polygons, so that any word could be completely contained at least in one polygon when a text page is placed in front of camera (see Fig. 8(b)). The question then arises, "which homography matrix should be chosen?" Given any word in the lowest-resolution text image, the homography matrix H_1, which can generate the complete bounding box of the word in one zoom-in image after transformation, is selected from the 18 homography matrices. If there are multiple ones which meet the requirement described above, they are in a queue for the consequent process.

C. Estimation of Homography II

After the transformation of homography I, a word in the lowest-resolution image is associated with the counterpart in the zoom-in image. When rotating PTZ camera to some positions where the zoom-in images need to be taken, perspective distortion may arise during the image projective transformation (see the example in Fig. 7). We need another homography to correct such distortion, otherwise it could jeopardize the accuracy of OCR. If there are four known vertices of a square in the image, it would be easy to compute such homography. Again, the chessboard image is used. In each zoom-in image that captures a partial view of the chessboard, the Harris corner detector is used for corner detection and the Hough transform is used for line detection. Then all squares in the image can be detected by combining these corners and lines. The square closest to the image center is selected and its vertices is used for homography estimation as follows:

$$p_b = H_2 p_a \tag{2}$$

where p_a is a vertex of the selected square and p_b is a vertex of the destination square. Since there are eight unknowns in the homography H_2, four points are enough to solve (2) by the least squares solution. Fig. 9 shows an example of perspective distortion correction by using homography transformation.

When both H_1 and H_2 are available, the matrix multiplication of H_1 and H_2 is given as:

$$H = H_2 H_1 \tag{3}$$

where H establishes a transformation between a low resolution word and its counterpart in the zoom-in and upright image. After the transformation by (3), if the bounding box of a word can completely appear in the zoom-in and distortion-free image, it is retained and forwarded to the next stage (see Fig. 7). Otherwise, the next H_1 in the list is selected to compute the H in (3).

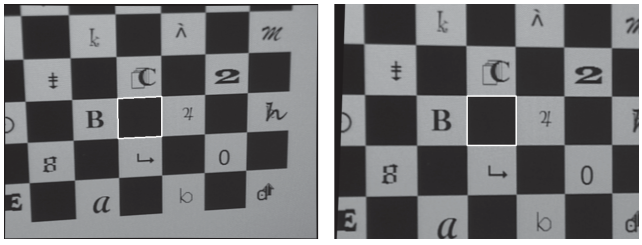


Fig. 9. Homography for perspective distortion correction. In the left image, the central square is selected for computing homography transformation. The corrected square is shown in the right image.

D. Output Layout and Tesseract OCR

Having obtained the high-resolution image of the words, the adaptive thresholding transforms gray image to binary form for suppressing the effect of large lighting variations on each word. These "enlarged" binary words should appear at appropriate locations in a larger output image, where the text layout is similar to one in the original image. First of all, the enlargement factor is computed by averaging factors of all transformed words with their original ones. In the original text image, the first word is considered as the anchor and the relative positions between it and other words are obtained in terms of the top-left corner of bounding box. Then in the output image, these relative positions are magnified according to the enlargement factor, and the corresponding binary words are placed at there. Fig. 7 shows an example of the layout-making process, where the output binary image has similar layout to the original gray image. This process can affect the eventual success or failure of implementing Tesseract-OCR [8] for the whole text.

Tesseract-OCR is considered as one of the most accurate open source OCR engines currently available. Tesseract handles input text image in grayscale or binary format and follows a traditional step-by-step pipeline for OCR. Refer to [8] for a detailed description.

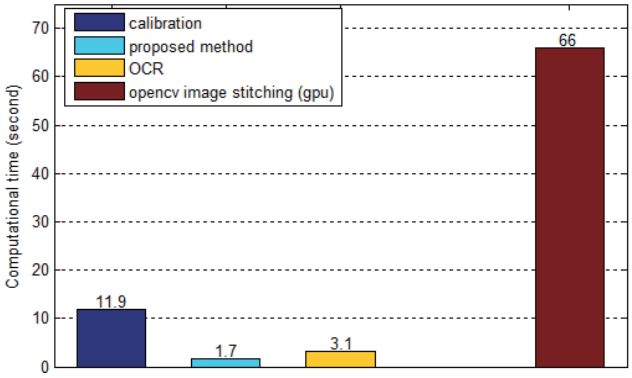
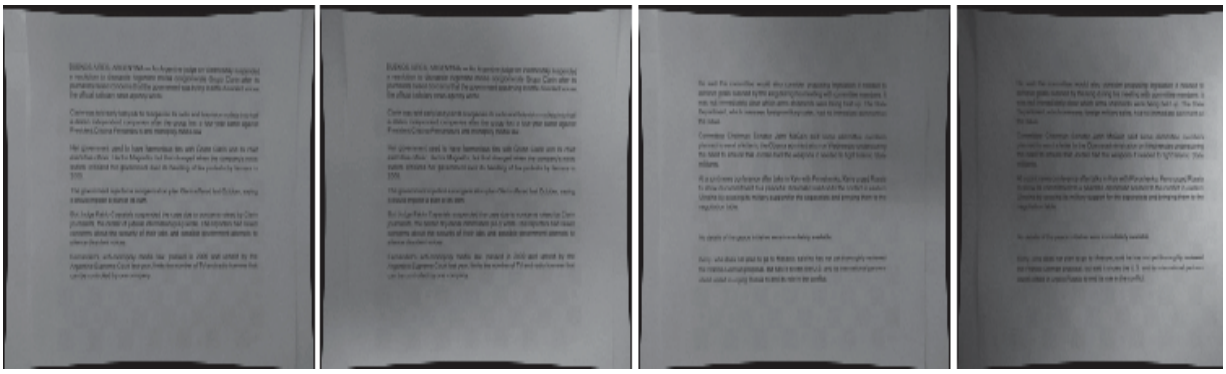


Fig. 10. Comparison of our method with [11] in terms of computational time.

E. Advantage of the proposed method

Our method can be considered as a kind of special image stitching for OCR, in the sense of combing multiple images to produce a high-resolution binary image. As the alternative, a well-known image stitching [11] can be used to produce a grayscale panorama for OCR. But this method is time consuming, as it relies on feature detection and matching in an online process to update a bunch of homographies and uses bundle adjustment for minimizing the global alignment error. In our method, feature detection and matching is used only in calibration that is an offline procedure. Also in OCR, binary image as input can achieve good recognition performance. Some procedures, which are implemented in [11] such as image blending for better image visualization, can be skipped in our method.

Fig. 10 shows the comparison of our method with image stitching in terms of computational time. Note that the method [11] is implemented in C++ and OpenCV with GPU, while our method is implemented in OpenCV-Python with CPU. The result shows our method is much faster than image stitching for OCR.



98.5% 97.5% 97.8% 94.5%

Fig. 11. Images of two text pages with different lighting conditions. The accuracy of word recognition is given below images.

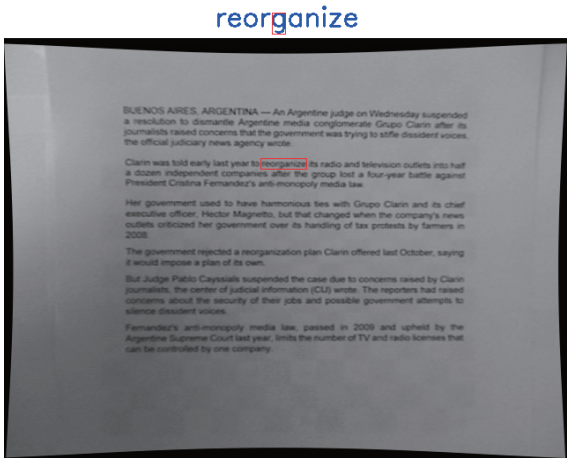


Fig. 12. OCR demo.

F. Issues of converting to finger-wearable system

Our current OCR system is based on the desk-mounted camera. The key point is establishing a whole-to-part mapping with the help of calibration using chessboard, so that any word in zoom-in image can be associated with its position in low-resolution but whole-text image. This significantly affects the output of OCR. When the system is transferred to a finger-wearable device, we have to know where the fingertip is on the page surface, or which portion the finger-wearable camera is looking at relative to the whole-text image. There might be two ways to reach the goal. In the first method, finger-wearable camera obtains the whole-text image in a distance to the page surface, and then the camera touches the page for obtaining the closer and clearer views. Building the mapping in this situation requires the more robust and reliable feature detection and matching. In the second method that are more complicated than the first one, a headset with 3D sensor, such as Intel RealSense, is used to obtain the page surface and fingertip in 3D camera space. A mapping between finger-wearable camera and 3D camera can be established, so that the field of view can be transferred from one to another. In the future, we will work on these methods to improve the OCR accuracy of Finger-Eye fitting to read the whole text.

III. Preliminary Experiments

The PTZ camera used is Sony EVI-D100. The zoom level is set to be 25 for zooming in and the size of image is 480*640. We have tested our OCR method on six text images acquired under different lighting conditions. Fig. 11 shows examples of two text pages and their accuracy of word recognition. It is evident that variation in illumination can affect the OCR accuracy. The average accuracy of our method is 97.5%. The method first performs OCR for whole text and then sends each recognized word to Raspberry Pi for generating e-Braille code every one second. Fig. 12 shows the picture of OCR demo window. The word being sent to Raspberry Pi is labeled in red rectangle in low-resolution image and its enlarged form is shown on top. The labeled letter in the word is being converted to E-Braille code.

IV. Conclusion

This paper presents a portable and refreshable text reading system, called Finger-eye. In this developed system, a small camera is added to the fingertip-electrode interface of the current Electro-tactile Braille Display and placed on a blind person's finger to continuously process images using a developed rapid optical character recognition (OCR) method. This will allow translation of text to braille or audio with natural movement as if they were reading any Braille Display or book. The braille system that will be used is a portable electrical-based braille system that will eliminate the problems associated with refreshable mechanical braille displays. The goal of the research is to aid the blind and visually impaired (BVI) with a portable means to translate any text to braille, whether in the digital realm or physically, on any surface.

Reference

[1] Braille Authority of North America. "Evolution of Braille". Brailleauthority.org, retrieved 25 February 2014.

[2] Loewen, G., and V. Tomassetti. "Fostering independence through refreshable Braille." Presentation at the Developing Skills for the New Economy: International Conference on Technical and Vocational Education and Training, Manitoba. 2002. Retrieved 25 February 2014.

[3] http://www.nbp.org/ic/nbp/technology/brailletechnology.html. Retrieved 4/30/2014.

[4] Gregory, John, Yantao Shen, and Ning Xi. "On-line bio-impedance identification of fingertip skin for enhancement of electrotactile based haptic rendering." Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on. IEEE, 2011.

[5] ANSI A117.1-2003 (2003), "American National Standard: Accessible and useable building and facilities," Chapter7. Communication Elements and Features: Section 703.4 Braille.

[6] Quick Start Guide. http://www.raspberrypi.org/wp-content/uploads/2012/04/quick-start-guide-v2_1.pdf. Retrieved 25 February 2014.

[7] Tesseract-OCR. https://code.google.com/p/tesseract-ocr/ Retrieved 25 February 2014.

[8] R. Smith, "An overview of the Tesseract OCR engine," 9th International Conf. Document Analysis and Recognition, 2007.

[9] H. Richard and A. Zisserman. Multiple View Geometry in Computer Vision, 2003.

[10] M. Muja and D. Lowe, "Fast approximate nearest neighbors with automatic algorithm configuration," VISAPP (1), 2009.

[11] M. Brown and D. Lowe, "Automatic panoramic image stitching using invariant features," International Journal of Computer Vision, 74(1), 2007